AN11802

NFC Reader Library for Linux Installation Guidelines

Rev. 1.4 — 16 May 2017
367914

Application note
COMPANY PUBLIC

Document information

Info	Content
Keywords	NFC, Reader Library, Linux, PN5180, CLRC663, Raspberry Pi
Abstract	This note describes how to install NXP’s NFC Reader Library on a GNU/Linux system

Revision history

Rev	Date	Description
1.4	20170516	Alignment on latest SW delivery v05.02.00 Updated reference platform
1.3	20170104	Alignment on latest SW delivery v4.050.03.001651
1.2	20161115	Alignment on latest SW delivery v4.040.05.011646 Updated examples descriptions
1.1	20160418	Corrected wrong connection between CLEV663B and Raspberry Pi
1.0	20160404	First release

Contact information

For more information, please visit: http://www.nxp.com

1. Introduction

The NFC Reader Library is a feature complete software support library for NXP's NFC Frontend ICs. It is designed to give developers a faster and simpler way to deliver NFC-enabled products. This multi-layer library, written in C, makes it easy to create NFC based applications. See [1] for more details.

The purpose of the present document is to give instructions on how to install the NFC Reader Library on a generic GNU/Linux platform.

It takes as reference the support of CLEV6630B board on Raspberry Pi platform. The reference environment is further described in chapter 2.

In below chapters, information highlighted thanks to surrounded borders relates to examples on this reference platform.

Finally it gives details about modifications to be done to make the porting of the NFC Reader Library for Linux to another Linux platform than the one used as reference, or for others NXP NFC Frontend ICs support.

2. Reference environment

2.1 Overview

The selected reference environment is the CLEV6630B v2.0 board (see [5]), including CLRC663 NXP’s NFC Frontend (see [4]), connected to Raspberry Pi platform (refer to [2] for more details) running Raspbian Jessie Linux distribution.

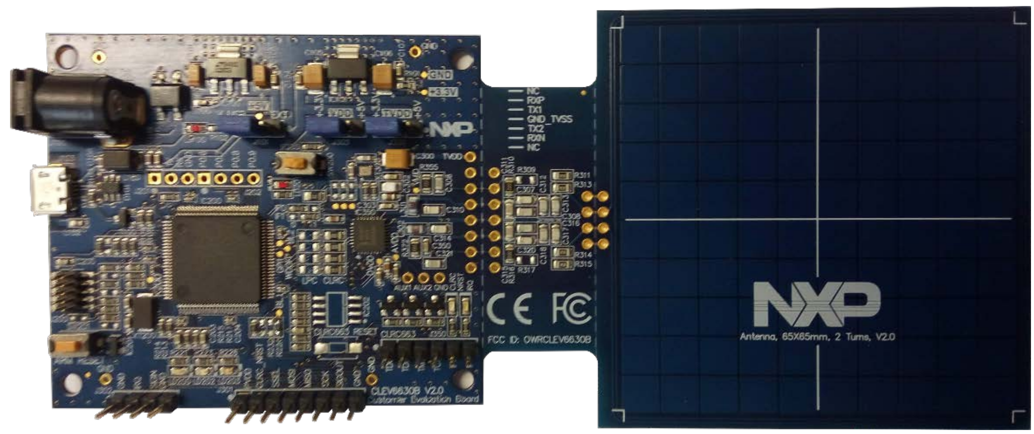

Fig 1. CLEV6630B v2.0 board

2.2 Boards connections

Using CLEV6630B board connection to the Raspberry Pi must be done according to Table 1 and **Error! Reference source not found.**.

The CLEV6630B board must have been previously set into the proper configuration as indicated in the related document chapter 6.

Table 1. CLEV663B connection to Raspberry Pi

Wiring to be done to connect CLEV6630B demo board to Raspberry Pi

CLEV6630B pin	Raspberry Pi	
	Pin number	Pin function
MOSI	19	SPI-MOSI (GPIO 10)
MISO	21	SPI-MISO (GPIO 9)
SCK	23	SPI-CLK (GPIO 11)
SSEL	24	SPI-CE0 (GPIO 8)
CLRC_NRST	18	GPIO 24
IRQ	16	GPIO 23
IFSEL0	13	GPIO 27
IFSEL1	15	GPIO 22
GND	6, 9, 14, 20, 25, 30, 34 or 39	GND

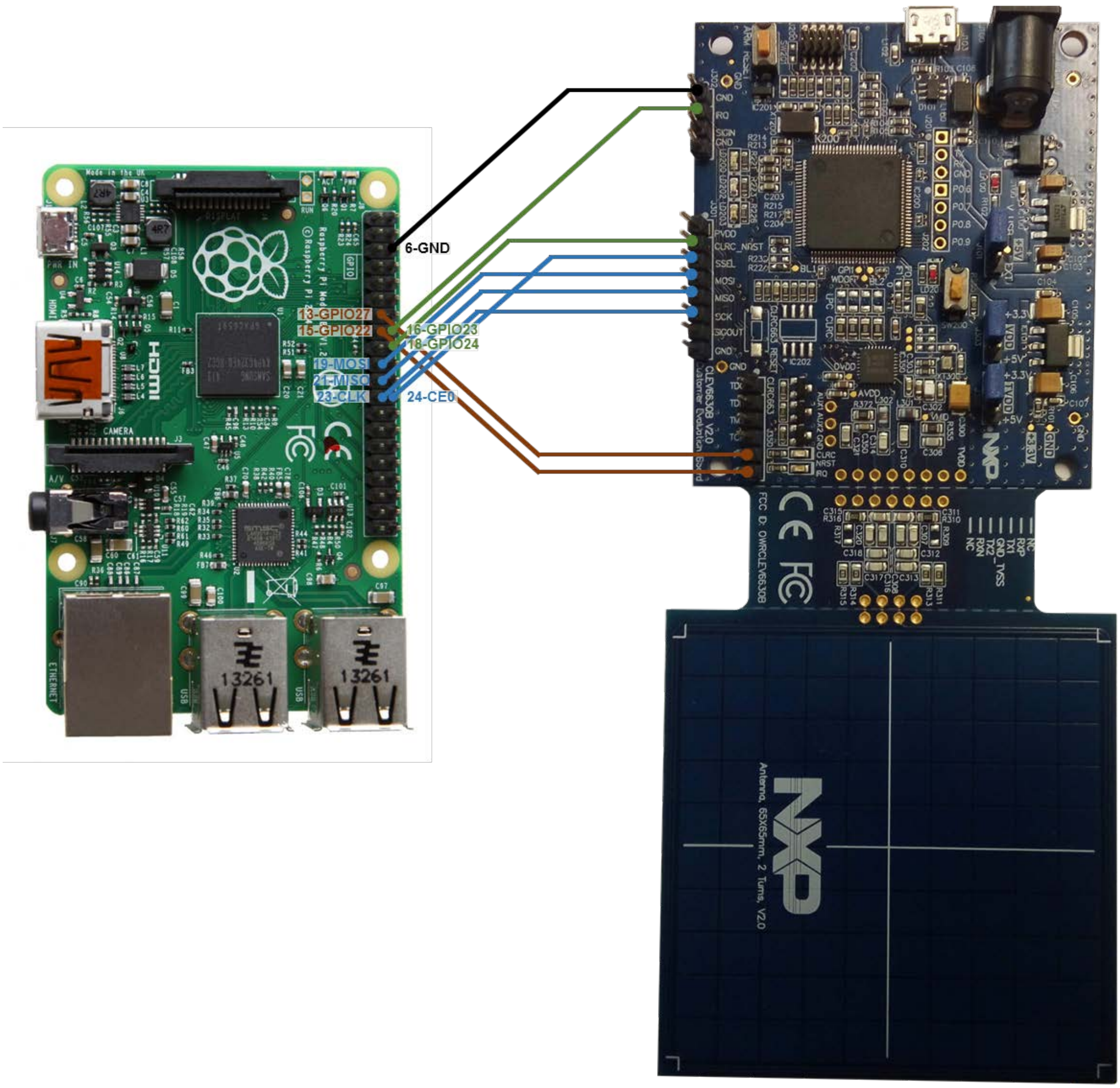

Fig 2. CLEV6630B connection to Raspberry Pi

3. NFC Reader Library for Linux delivery

3.1 Step 1: installing NFC Reader Library for Linux delivery

The NFC Reader Library delivery consists of a zip file to be uncompressed on the Linux target (or on host machine in case of cross compilation):

```
$ unzip sw3693.zip
```

This creates the following directory structure in the current path:

```
├── docs
│   └── NxpNfcRdLib
│       └── NxpNfcRdLib
│       └── DAL
│       └── OSAL
├── Examples
│   └── NfcrdlibEx1_BasicDiscoveryLoop
│   └── NfcrdlibEx2_AdvancedDiscoveryLoop
│   └── NfcrdlibEx3_NFCForum
│   └── NfcrdlibEx4_MIFAREClassic
│   └── NfcrdlibEx5_ISO15693
│   └── NfcrdlibEx7_EMVCo_Polling
│   └── NfcrdlibEx8_HCE_T4T
│   └── NfcrdlibEx9_NTagI2C
│   └── Nfcrdlib_SimplifiedAPI_ISO
├── ComplianceApp
│   └── NfcrdlibEx11_ISO10373_PCD
│   └── Nfcrdlib_SimplifiedAPI_EMVCo
│   └── Nfcrdlib_SimplifiedAPI_EMVCo_Analog
├── NxpNfcRdLib
├── Platform
├── RTOS
├── CMakeLists.txt
└── NXP_SLDA.pdf
└── README.txt
```

- {docs} contains the NxpNfcRdLib as well as DAL and OSAL layers Doxygen documentation, entry points being the html files
- {Examples} contains code examples of the NFC Reader Library use (refer to chapter 5)
- {ComplainceApp} contains code of applications for compliancy testing (refer to chapter 5)
- {NxpNfcRdLib} is the NFC Reader Library source code
- {Platform} is the DAL layer source code (relates to RaspberryPi)

- {RTOS} is the OSAL layer source code (relates to Linux)
- {CMakeLists.txt} allows building makefiles structure
- {NXP_SLDA.pdf} describes the NXP Software License and Distribution Agreement of the current package.
- {README.txt} gives information about the NFC Reader Library for Linux delivery

3.2 Step 2: generating makefiles structure

Generating the makefiles structure requires the following modules been installed on the target (or on host machine in case of cross compilation):

- *gcc*
- *make*
- *cmake* (version 2.8.11 minimum)

On reference platform (Raspberry Pi running Raspbian Jessie), *cmake* installation is done using command:

```
$ sudo apt-get install cmake
```

Makefiles generation is then done running *cmake* command, from the build sub-directory:

```
$ mkdir _build
$ cd _build
$ cmake ..
```

3.3 Step 3: enabling the SPI physical interface

The physical link used to interface the NFC Frontend must be enabled on the platform: node */dev/spidev* must be present and accessible.

On the Raspberry Pi, enabling SPI is done through raspi-config tool:

```
$ sudo raspi-config
```

The option to activate SPI can be found in:
Advanced Options → SPI → <Yes>.
Then reboot is required to take into account the change:

```
$ sudo reboot
```

3.4 Step 4: building and running the examples

Building the examples (including the NFC Reader Library source code) is then just done with the simple *make* command from the *_build* directory:

```
$ cd ../_build
$ make all
```

Examples can also be individually built from the dedicated sub-directory:

```
$ cd ../_build/Examples/NfcrdlibEx<#_ExampleName>
$ make
```

This generates `NfcrdlibEx<#_ExampleName>` example binary in the current directory.

Running the example (complete description of examples in chapter 5) is done executing the related command:

```
$ ./NfcrdlibEx<#_ExampleName>
```

Below is the output obtained running *NfcrdlibEx1_BasicDiscoveryLoop*, tapping an NFC tag:

```
$ ./NfcrdlibEx1_BasicDiscoveryLoop
 BasicDiscoveryLoop Example:


 Card detected and activated successfully...
  Technology  : Type A
     Card: 1
     UID : 04 60 32 6A 64 34 80
     SAK : 0x00
     Type: Type 2 Tag
```

4. Porting of the NFC Reader Library for Linux

By default, the NFC Reader Library for Linux delivery is suitable to run the reference platform (described in chapter 2). In case of different setup, some adaptations are required.

4.1 Support of CLRC663

The way CLRC663 is connected is defined in *Board_PiRc663.h* file located under *Platform/DAL/boards* sub-folder of the delivery. In case of different connection than the one depicted at chapter 2, this has to be reflected there.

4.2 Support of PN5180

Selection of the NFC Frontend IC is done when generating the makefiles structure (see Step 2 described at chapter 3.2).

Selecting PN5180 (refer to [3] for more information about PN5180) is done defining adding `-DFRONTEND_PN5180=ON` parameter to the cmake command, additionally parameter `-DFRONTEND_RC663=OFF` must also be defined to unselect RC663 use (only one frontend can be selected at a time).

Using the PNEV5180B board connection to the Raspberry Pi must be done according to Table 2 and Fig 3.

The PNEV5180B board must have been previously set into the proper configuration as indicated in the related document chapter 6.

Table 2. PNEV5180 connection to Raspberry Pi
Wiring to be done to connect PNEV5180 demo board to Raspberry Pi

PNEV5180 pin	Raspberry Pi	
	Pin number	Pin function
MOSI	19	SPI-MOSI (GPIO 10)
MISO	21	SPI-MISO (GPIO 9)
SCK	23	SPI-CLK (GPIO 11)
NSS/SSEL	24	SPI-CE0 (GPIO 8)
nRESET/PN_RST	26	GPIO 7
BUSY	22	GPIO 25
IRQ	16	GPIO 23
DWL	6, 9, 14, 20, 25, 30, 34 or 39	GND
GND	6, 9, 14, 20, 25, 30, 34 or 39	GND

In case different connection is defined, this must be reflected in *Board_PiPn5180.h* file located under *Platform/DAL/boards* sub-folder of the delivery.

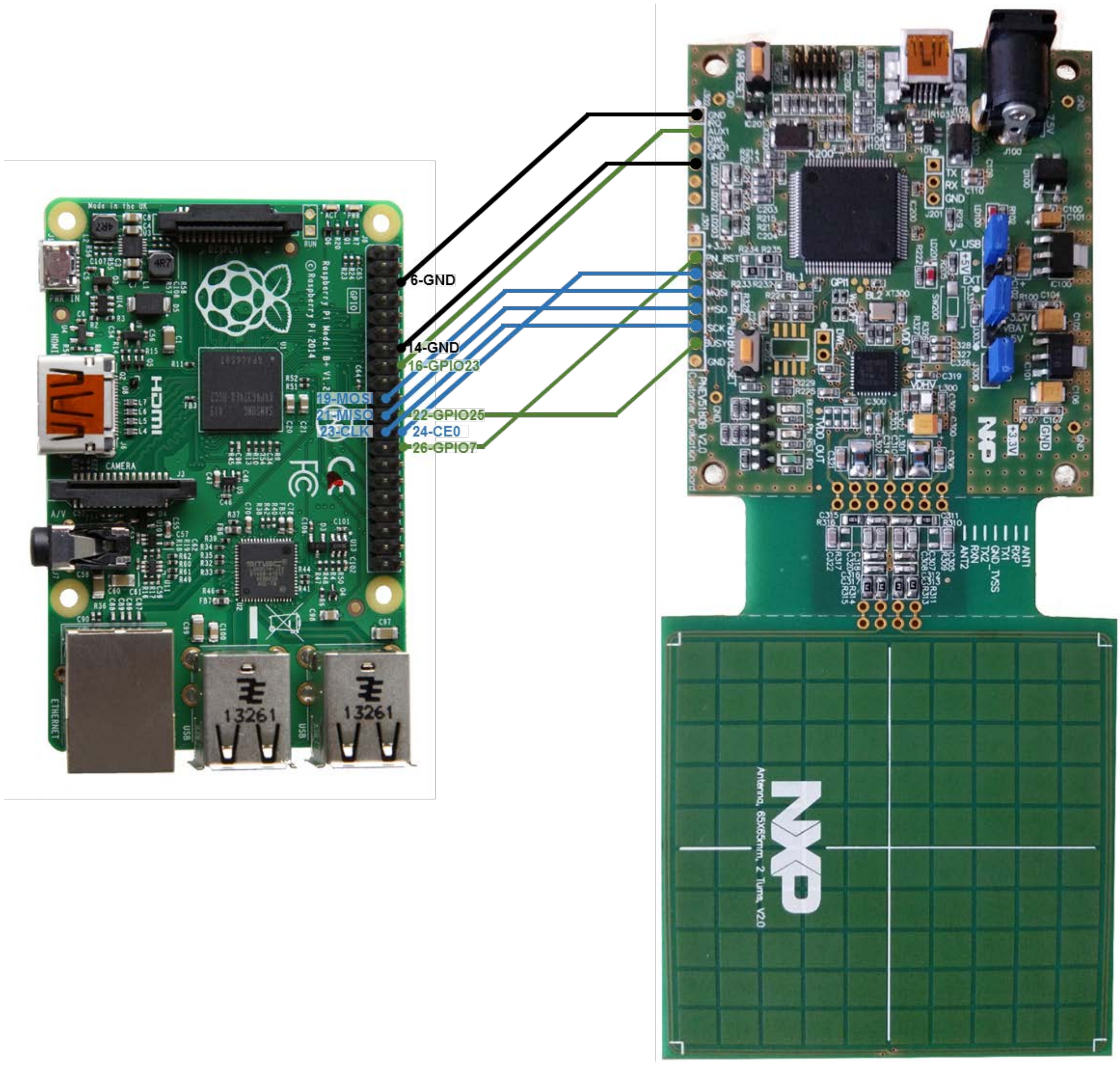

Fig 3. PNEV5180 connection to Raspberry Pi

4.3 Support of "bal" kernel module

!!! Only PN5180 is currently supported by the current "bal" kernel module!!!

To insure critical timings (for instance for EMVco compliancy) on slow platforms, use of "bal" kernel module is recommended. It abstracts SPI access which significantly speed-up the communication time between the Linux platform and the NFC Frontend IC.

Source code of the “bal” kernel module is provided on the following GitHub repository: https://github.com/NXPNFCLinux/nxprdlib-kernel-bal. All information to build and install this kernel module on the Linux platform is described in the related *README.md* file present in this repository.

When properly loaded on the Linux platform, the module should expose */dev/bal* device node.

To make use of this kernel module related **`PHDRIVER_LINUX_KERNEL_SPI`** definition must be uncommented inside *Board_PiPn5180.h*, located under *Platform/DAL/boards* sub-folder of the delivery. Other definition **`PHDRIVER_LINUX_USER_SPI`** must be commented out.

5. NFC Reader Library examples

5.1 Examples

The NFC Reader Library is delivered together with examples demonstrating functionality and use of the API.

5.1.1 Example #1 – Basic Discovery Loop

The Discovery Loop can be seen as the entry point when starting to communicate with an NFC tag or device. It scans the close environment for tags and devices of different technologies. Example is implemented to work in POLL and LISTEN mode of the discovery loop. Information (like UID, SAK, and Product Type for MIFARE Cards) of the detected tags are printed out and it also prints information when it gets activated as a target by an external initiator/reader. Whenever multiple technologies are detected, example select first detected technology and resolve it.

In passive poll mode, Low Power Card Detection (LPCD) is enabled.

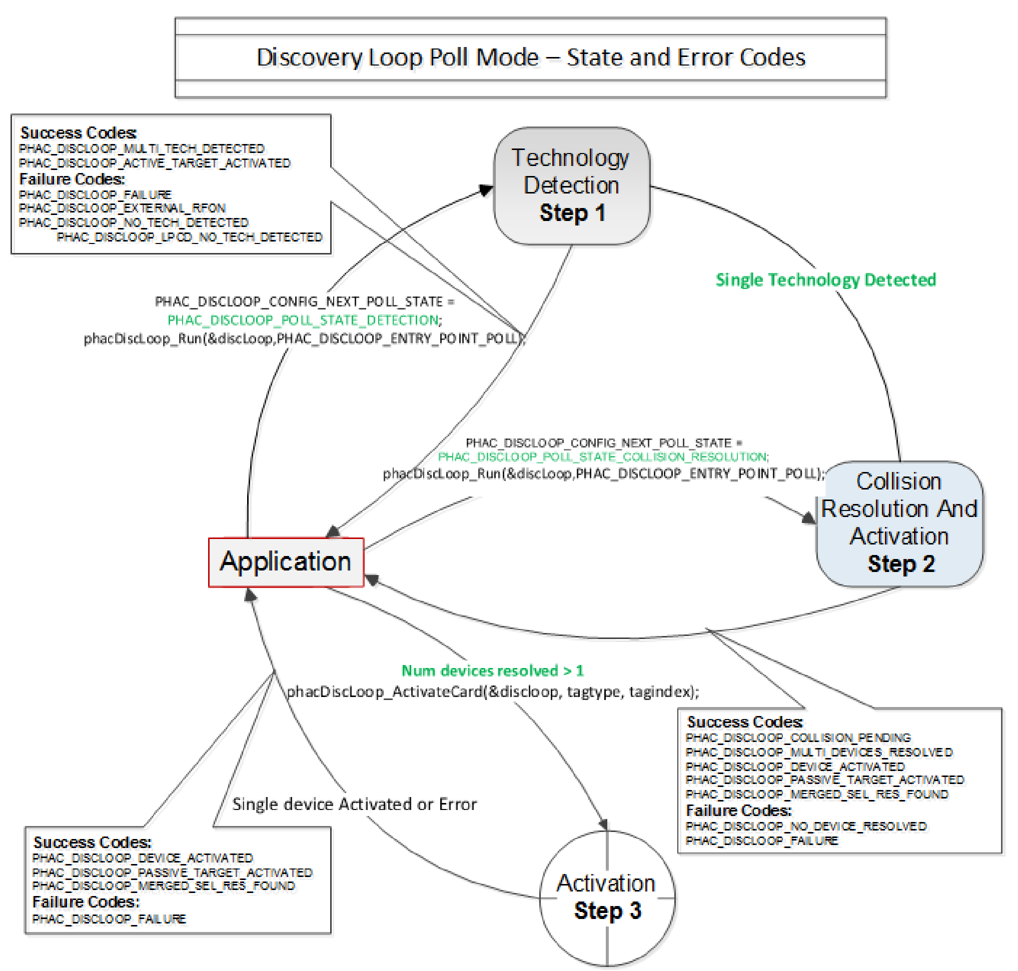

Fig 4. Discovery Loop in Poll mode

The core function of this example is "*BasicDiscoveryLoop_Demo()",* where initialization of the NFC Reader library and polling for NFC technologies is implemented. After each polling loop, application is checking polling result and printout information about the detected tags or devices.

This example is using default DiscoveryLoop configuration, which enables all supported technologies and it is limited to one device for each technology.

Table 3. Supported technologies

ISO14443P3A	ISO15693- SLI	FeliCa	TYPEF_TARGET_PASSIVE
ISO14443P4A	ISO18000P3M3	TYPEA_TARGET_PASSIVE	TYPEF_TARGET_ACTIVE
ISO18092MPI	ISO14443P3B	TYPEA_TARGET_ACTIVE	

5.1.2 Example #2 – Advanced Discovery Loop

Additionally to Example 1 the Advanced Discovery Loop example explains the different configuration options of the Discovery Loop and configure DiscoveryLoop with default values based on the interested profile, NFC or EMVCo.

The configuration of the "DiscoveryLoop" is implemented in "*LoadProfile()*" function.

5.1.3 Example #3 – NFC Forum

Explains how to configure the NFC Reader Library for different P2P modes such as Active Mode, Target Mode, Initiator Mode and SNEP Client/Server.

In Snep Server mode the example waits for a connection from a Snep Client. When the connection between client and server is establish, client send a data and server read it. The application displays read data in the console window of the LPCXpresso IDE.

In Snep Client mode, the application tries to connect to a Snep Server. Once the connection is established, it transmits an NDEF message to the server.

> !!! CLRC663 not supporting NFC Target Mode feature, this example cannot be built for the related configuration!!!

5.1.4 Example #4 – MIFARE Classic

This example demonstrate how to configure "DiscoveryLoop" to poll for only one technology and how to resolve detected card, in this example MIFARE Classic is used.

Once MIFARE Classic card is activated, application printout information like UID, ATQA and SAK and perform the authentication with MIFARE default key. After successful authentication basic read/write operations are implemented.

This example is good start in case of working with only one card or to see how to manage MIFARE Classic cards.

5.1.5 Example #5 - ISO15693

Similar to the previous example, this one is also using only one technology, in that case ISO15693. “*DiscoveryLoop*” is configured to resolve only one device and in the example it is shown how to change Tx Guard Time for T5T cards, this is implemented in “phApp_Init()” function.

Once ICODE SLI is resolved and activated, application printout card information like type of the card and UID, and it will read and write from/to the memory block.

This example is good start in case of working with only one card or to see how to manage ISO15693 type of the cards.

5.1.6 Example #7 – EMVCo Polling

The EMVCo Polling example it is demonstrated how to configure NFC Reader Library as specified by EMVCo specifications and starts polling for EMVCo cards.

Once an EMVCo compatible card is resolved and activated, it demonstrates the exchange of APDU commands. This example shall help the developers getting started more quickly when working with EMVCo cards.

5.1.7 Example #8 – HCE T4T

Example 8 implements a Type 4 Tag card emulation according to NFC Forum Type 4 Tag specification. The example supports all specified commands such as *Select*, *ReadBinary*, *UpdateBinary*.

With this example our reader is in card emulation mode (HCE) and it support reading and writing data. Default data is configured as an NDEF message as a url www.nxp.com.

The maximum NDEF length the reader can write is limited by NDEF file size used in example (default configured as 1024 bytes).

!!! CLRC663 not supporting Card Emulation feature, this example cannot be built for the related configuration !!!

5.1.8 Example #9 – NTAG-I2C

The NTAG-I2C example demonstrates the use of special features which are supported by NTAG-I2C. By using POLL mode of the discovery loop, example detect the NTag I2C cards and displays detected tag information like UID, ATQA, SAK, Version info and perform “*Page Read*” and “*PageWrite*” commands.

For more details about the NTAG-I2C and its functionalities please consult the related product page.

5.1.9 SimplifiedAPI ISO

This example is a reference application to demonstrate the usage of Simplified API with ISO profile. Application contains example of Type A Layer 4, Type B Layer 4, MIFARE Desfire, MIFARE Ultralight, MIFARE Classic, ISO5693 and ISO18000p3m3.

Example demonstrates how to use simplified API, which require, after successful library initialization, only three commands:

- *phNfcLib_Activate()*
- *phNfcLib_Transmit()*
- *phNfcLib_Receive()*

5.2 Compliancy applications

The NFC Reader Library is also delivered together with applications allowing to execute compliancy tests.

5.2.1 Example #11 – ISO10373

This example is an Application which is used to perform ISO 10373-6 PCD compliance validation. This example has to be executed in the DUT which has an ISO 14443 based PCD implementation. The ISO 10373-6 test methods verifies the compliance to the ISO 14443 protocols. An external tool like Micropross MP300 implements the test methods for the ISO 10373-6 and is used as the counterpart for this testing.

5.2.2 SimplifiedAPI EMVCo

This example is a LoopBack Application which is used to perform EMVCo2.6(L1) digital compliance validation. The CPU frequency configured for the Host controller platforms should be sufficient enough to meet EMVCo timing requirements.

5.2.3 SimplifiedAPI EMVCo Analog

This example contains three mode of operations within itself for the user to choose as below:

1. EMVCo LoopBack Application
2. Trans send Type A application
3. Trans send Type B application

Above Application modes are used to perform EMVCo2.6(L1) Analog compliance validation. The CPU frequency configured for the Host controller platforms should be sufficient enough to meet EMVCo timing requirements.

6. Annex 1: CLEV6630B rework for direct access

In order to assure direct access to the CLRC663 frontend IC (bypassing LPC1769) on the CLEV6630B board, the following rework of the CLEV6630B v2.0 has to be done:

In total six resistors (R359/R360 and R362/R364/R366/R368) in two groups (interface selection and connection to LPC1769) need to be removed to obtain proper decupling of the LPC1769 MCU from the CLEV6630B board. See on Fig 6 resistors to be removed marked red.

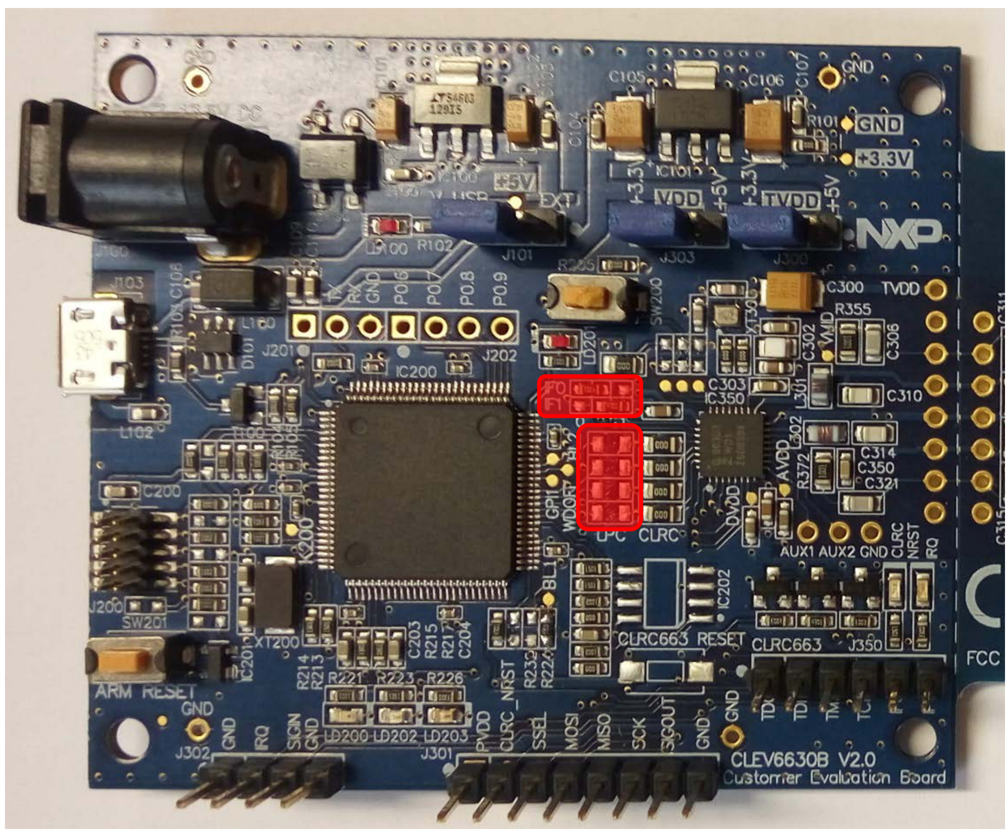

Fig 5. CLEV663 board rework

To check for the correct CLEV663B board configuration please refer to [5].

The Power of CLEV663B board still come from either the USB or the 5V Power connectors.

7. Annex 2: PNEV5180B rework for direct access

In order to assure direct access to the PN5180 frontend IC (bypassing LPC1769) on the PNEV5180B board, the following rework of the PNEV5180B v2.0 has to be performed:

In total six resistor in two groups need to be removed to obtain proper decupling of the LPC1769 MCU from the PNEV5180 board. See on Fig 6 resistors to be removed marked red.

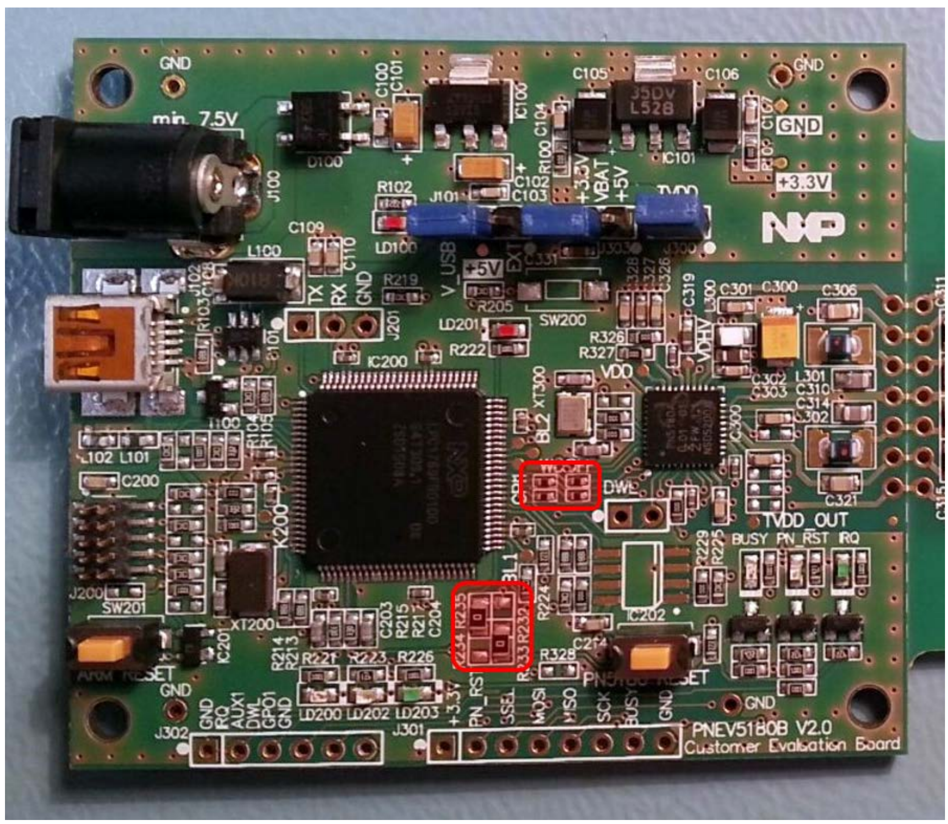

Fig 6. PNEV5180B board rework

To check for the correct PNEV5180B board configuration please refer to [6].

The Power of PNEV5180 board still come from either the USB or the 5V Power connectors.

8. References

[1] The NFC Reader Library gives Software support for NFC Frontend solutions.

For more information about it please visit http://www.nxp.com/products/:NFC-READER-LIBRARY.

[2] The Raspberry Pi is a credit card sized computer. To get started quickly, the Raspberry Pi Foundation provides several preconfigured Linux distributions.

For more information about it please visit http://www.raspberrypi.org/

[3] PN5180 is a highly integrated high performance full NFC Forum-compliant frontend IC for contactless communication at 13.56 MHz.

For more information about it please visit http://www.nxp.com/products/:PN5180.

[4] CLRC663 *plus*, the high performance multi-protocol NFC frontend.

For more information about it please visit http://www.nxp.com/products/:CLRC66303HN

[5] AN11211 Quick Start Up Guide RC663 Blueboard.

[6] UM10954 PN5180 SW Quick start guide.

9. Legal information

9.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

9.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

9.3 Licenses

Purchase of NXP ICs with NFC technology
Purchase of an NXP Semiconductors IC that complies with one of the Near Field Communication (NFC) standards ISO/IEC 18092 and ISO/IEC 21481 does not convey an implied license under any patent right infringed by implementation of any of those standards. Purchase of NXP Semiconductors IC does not include a license to any NXP patent (or other IP right) covering combinations of those products with other products, whether hardware or software.

9.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

MIFARE — is a trademark of NXP B.V.

10. List of figures

11. Contents

Date of release: 16 May 2017
367914
Document identifier: AN11802